Framework 2
Manual de usuario del Servicio de envío de SMS

Versión 1.0

Área de Aplicaciones Especiales y
Arquitectura de Software

Agencia de Informática y Comunicaciones de la Comunidad de Madrid

Hoja de Control

Título	Framework 2: Servicio de envío de SMS		
Documento de Referencia			
Responsable	Área de Aplicaciones Especiales y Arquitectura de Software		
Versión	1.0	**Fecha Versión**	

Registro de Cambios

Versión	Causa del Cambio	Responsable del Cambio	Fecha del Cambio
1.0	Versión inicial del documento	Area de Aplicaciones Especiales y Arquitectura de Software	
1.1	Inclusión de nuevos métodos en la interfaz del servicio.	Área de Arquitectura de Aplicaciones	14/04/2015

Contenido

1. INTRODUCCIÓN 4

1.1. AUDIENCIA OBJETIVO 4
1.2. CONOCIMIENTOS PREVIOS 4

2. DESCRIPCIÓN 5

enviaSMSxml 8

3. USO DE LA LIBRERÍA MENT_LIB 9

3.1. CLASES DE PARAMETROS TIPO OBJETOS 10
3.1.1. SmsSimpleBean 10
3.1.2. SmsDetalladoBean 10
3.1.3. ConsultaPeticionBean 11
3.1.4. ConsultaDetalleXMLBean 12
3.1.5. RespuestaDetalleXmlBean 12
3.2. METODOS CON PARAMETRO TIPO OBJETO 14
3.2.1. enviaSMSSimples 14
3.2.2. enviaSMSDetallados 15
3.2.3. consultaPeticion 17
3.2.1. consultaDetallePeticion 18
3.2.2. consultaDetallePeticionTime 19
3.2.3. consultaSMSxml 20
3.1. ESQUEMAS PARA LOS FICHEROS XML 23
3.1.1. XML de envío 23
3.1.2. XML de consulta 25
3.2. METODOS CON PARAMETRO TIPO STRING QUE CONTIENE UN XML 26
3.2.1. enviaForms 26
3.2.2. enviaSMSxml 27
3.2.3. consultaPeticionForms 29
3.2.4. consultaDetallePeticionForms 29
3.2.5. consultaSMSxmlForms 30
3.2.6. consultaSMSxmlForms 31

4. URL DE ACCESO AL SERVICIO WEB 33

4.1. DESARROLLO 33

5. METODOS Y PARAMETROS DEPRECADOS 34

5.1. ENVIOBEAN 34
5.2. LISTAENVIOSBEAN 35
5.3. RESPUESTABEAN 35
5.4. SMSBEAN 35
5.5. CONSULTAENVIOS 35
5.6. ENVIAMENSAJES 36

6. PREGUNTAS MÁS FRECUENTES 37

1. INTRODUCCIÓN

Para que las aplicaciones de la Comunidad de Madrid puedan enviar SMS se ha desarrollado un servicio web que será invocado desde las aplicaciones que necesiten implementar este requisito. De esta forma toda la complejidad del envío de SMS queda encapsulada en una llamada a un servicio web.

Dentro del framework 2 se ha incluido una librería que es una librería cliente de estos servicios web de forma que las aplicaciones del framework 2 que necesiten enviar sms deberán utilizar esta librería.

La librería se llama ment_lib y se encuentra publicada en la web de soja en el apartado Framework 2.

1.1. AUDIENCIA OBJETIVO

Este documento está orientado a aquellos desarrolladores de aplicaciones con el framework 2 que requieran implementar envío de sms en sus aplicaciones.

1.2. CONOCIMIENTOS PREVIOS

Para un completo entendimiento del documento, el lector deberá tener conocimientos previos sobre las siguientes tecnologías:

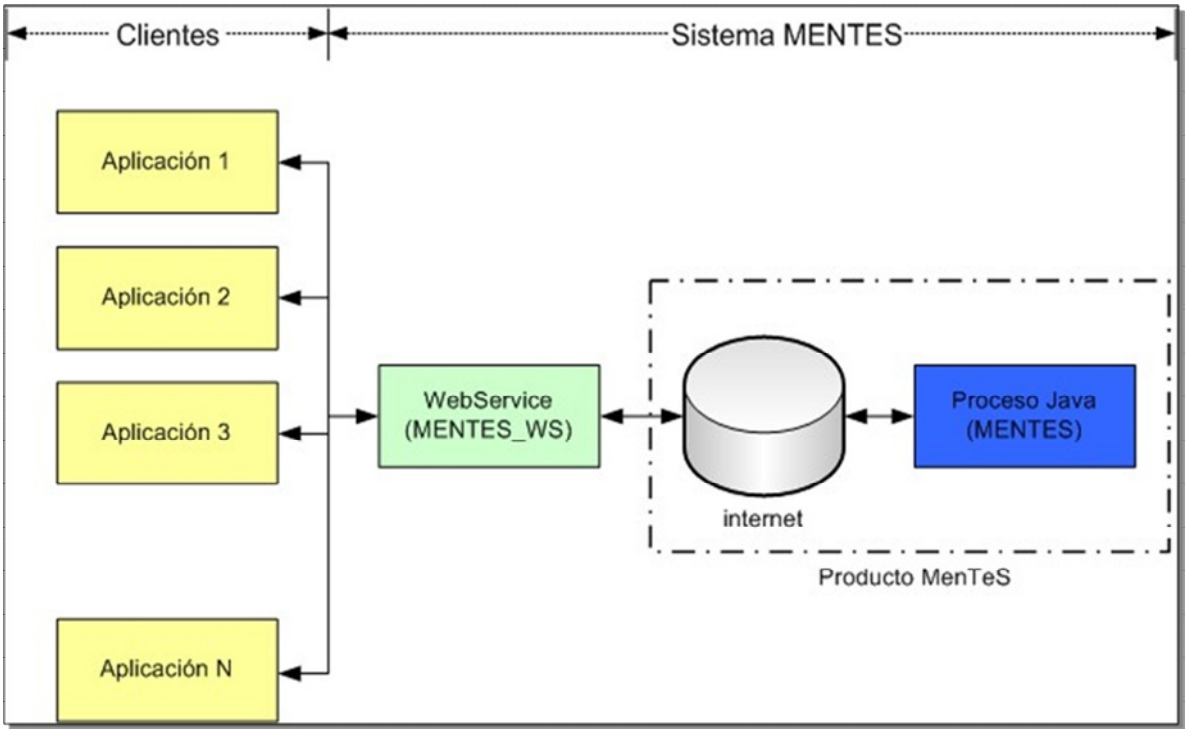
- Servicios Web
- WSDL
- Java

2. DESCRIPCIÓN

El servicio de envío de sms incluye las siguientes funcionalidades:

- Envió de 1 a n mensajes de texto a 1 o varios números de teléfono.
- Envío en el momento de la solicitud. (Hay que tener en cuenta que el servicio web encola las peticiones en la plataforma mentes por lo que si hay varias peticiones concurrentes los mensajes no son enviados en el mismo momento).
- Envío en diferido a una fecha/hora en concreto.
- Envío dentro de una ventana de tiempo tanto dentro de determinadas horas como de determinados días.
- Tiempo de expiración del mensaje en caso de que no se pueda enviar.
- Confirmación de envío (solamente con algunos operadores).
- Consulta de mensajes enviados.

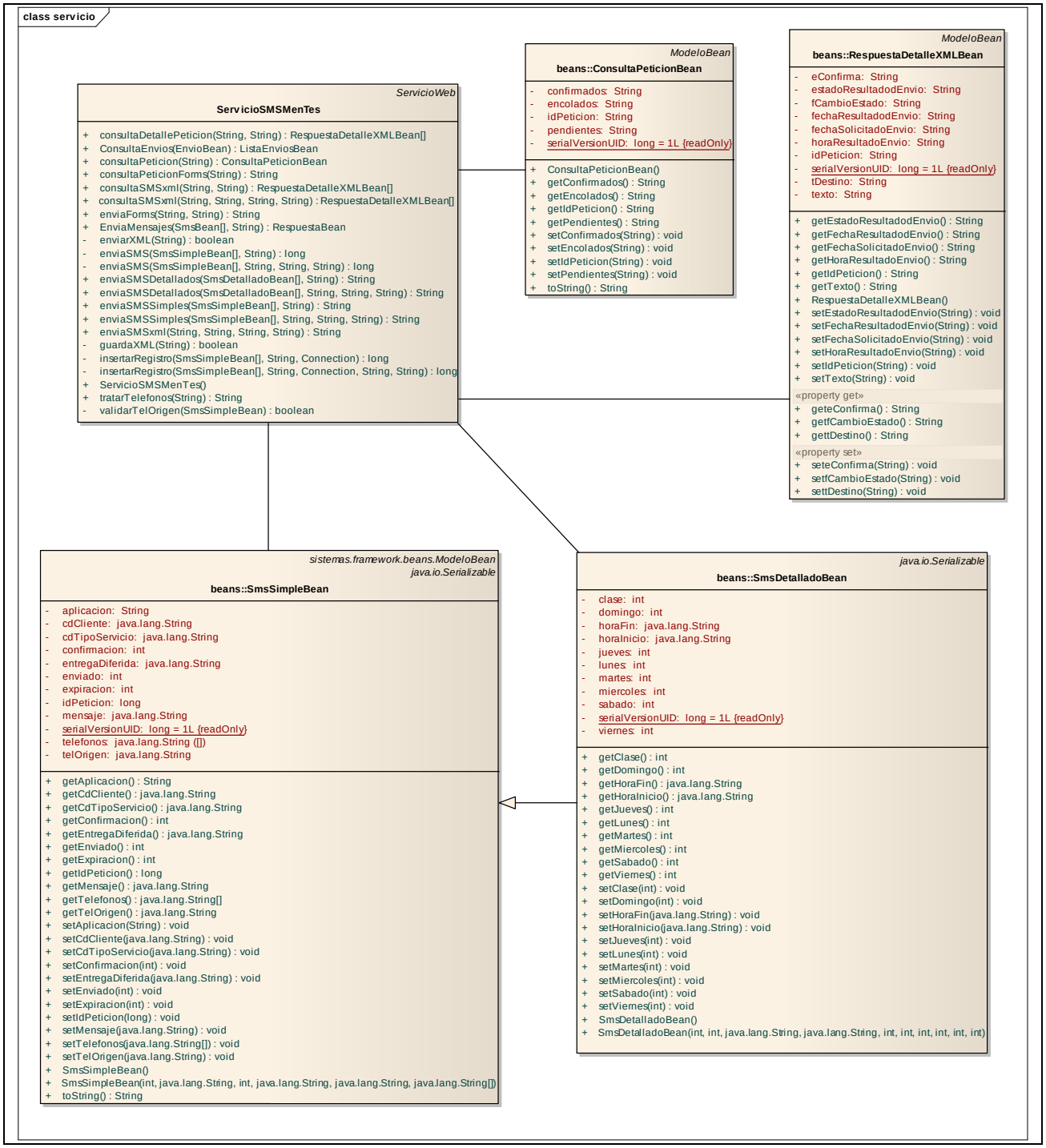
El servicio de envío de sms se ha implementado en un servicio web llamado **mentes_ws**. Este servicio web se comunica con la **plataforma MenTeS de Telefónica Soluciones** que en realidad es la encargada de la transmisión de los mensajes a los móviles.



La interfaz que define el servicio web se encuentra en el fichero **mentes_ws.wsdl** y se llama **ServicioSMSMentes**.

Las aplicaciones que necesiten acceder a este servicio web deberán crear un cliente de este servicio web partiendo del fichero mentes_ws.wsdl.

A continuación se muestra un diagrama de clases del servicio web



La forma de pasar parámetros a los métodos del servicio puede ser de dos formas:

- Mediante objetos
- En formato XML (En atributos de tipo String)

A continuación se muestra los distintos objetos que se van a utilizar como parámetros de los métodos del servicio ServicioSMSMentes.

Objeto	Descripción
SmsSimpleBean	Objeto con los datos básicos de una solicitud de envío de sms
SmsDetalladoBean	Objeto con los datos básicos para el envío de sms más los datos de la ventana de tiempo.
ConsultaPeticionBean	Objeto que se rellena con los datos del estado de un envío
ListaEnviosBean	Objeto que se que se rellena con una lista de objetos del tipo EnvioBean
RespuestaDetalleXMLBean	Objeto que devuelve la situación detallada de los SMS solicitados
ConsultaDetalleXMLBean	Objeto que se rellena con el código de aplicación,tipo de servicio y cliente,así como la fecha de envio.
XML de envío (String)	Xml que contiene una solicitud de envío de sms de acuerdo al esquema **envio.xsd.**
XML de consulta (String)	Xml que contiene la información del estado de una petición de envío de acuerdo al esquema **consulta.xsd**

La siguiente tabla muestra la lista de métodos que ofrece el servicio web.

	Métodos	Descripción
Métodos con parámetros de tipo objeto	**enviaSMSSimples**	Envío de SMS de manera simplificada, sin opciones de ventana de envío. Devuelve un código de petición para su posterior consulta.
	enviaSMSDetallados	Envío de SMS con opciones de ventana de envío. Devuelve un código de petición para su posterior consulta.
	consultaPeticion	Método de consulta de estado de una petición. Devuelve un objeto de tipo ConsultaPeticionBean con el estado de los SMS solicitados.
	consultaDetallePeticion	Método de consulta de la situación detallada de cada uno de los SMS solicitados para una petición concreta.
	consultaDetallePeticionTime	Método de consulta de la situación detallada de cada uno de los SMS solicitados para una petición concreta, este método incluye la información de horas minutos y segundos en la fecha de cambio de estado.
	consultaSMSxml	Método de consulta de la situación detallada de cada uno de los SMS solicitados en una fecha concreta para una aplicación, o aplicación, centro y servicio.

	ConsultaEnvios	Método deprecated,se mantiene por compatibilidad con los clientes del webservice ssms_ws. Consulta por fecha , aplicación, número de mensajes solicitados y número de mensajes enviados.Devuelve un objeto del tipo ListaEnviosBean
	EnviaMensajes	Método deprecated,se mantiene por compatibilidad con los clientes del webservice ssms_ws. Envia mensajes.Devuelve un objeto del tipo RespuestaBean,con el numero de mensajes enviados sus códigos de petición para posteriores consultas,ó en caso de error ,con el código de error.
Métodos con parámetros tipo xml	**enviaForms**	Envío de sms en el que la petición viene definida en un xml.
	enviaSMSxml	Envío de sms en el que la petición viene definida en un xml. Con parámetros: centro y tipo de servicio.
	consultaPeticionForms	Método de consulta de estado de una petición. Devuelve un string con el estado de los SMS solicitados con el siguiente formato: C=nº confirmados;P=nº pendientes;E=nº encolados.
	consultaDetallePeticionForms	Método de consulta,devuelve la situación detallada de cada uno de los SMS solicitados para una petición concreta,y una aplicación concreta,en formato xml
	consultaSMSxmlForms	Método de consulta,devuelve la situación detallada de cada uno de los SMS solicitados en una fecha concreta para una aplicación,en formato xml
	consultaSMSxmlForms	Método de consulta,devuelve la situación detallada de cada uno de los SMS solicitados en una fecha concreta para una aplicación ,centro y servicio,en formato xml

Es necesario que las aplicaciones se identifiquen en el servicio web con un **código único de aplicación** que permita el obtener estadísticas de uso por parte de las aplicaciones. Es necesario por lo tanto solicitar a ICM este código de aplicación antes de la puesta en producción del aplicativo.

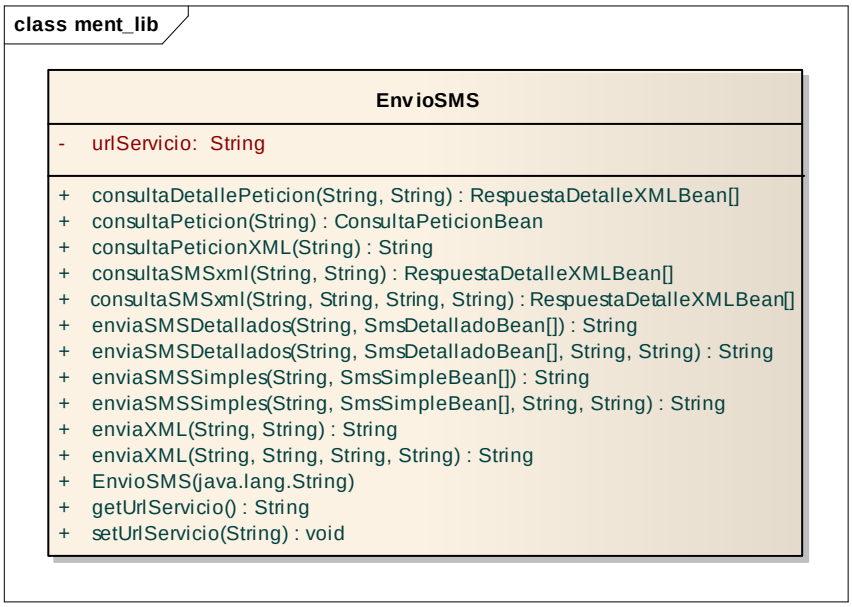
Se ha desarrollado una librería para que las aplicaciones Java no tengan que crear cada vez el cliente del servicio

web. Esta librería se llama **ment_lib**.

En el caso de que la aplicación sea con otro lenguaje distinto de java será la aplicación la encargada de crear el cliente del servicio web a partir del wsdl.

3. USO DE LA LIBRERÍA MENT_LIB

La librería **ment_lib** incluye la clase **EnvioSMS** que incluye los mismos métodos y parámetros del **ServicioSMSMentes** ya descrito anteriormente, por lo tanto no se vuelven a describir estos métodos.


class ment_lib
EnvioSMS
- urlServicio: String
+ consultaDetallePeticion(String, String) : RespuestaDetalleXMLBean[]
+ consultaPeticion(String) : ConsultaPeticionBean
+ consultaPeticionXML(String) : String
+ consultaSMSxml(String, String) : RespuestaDetalleXMLBean[]
+ consultaSMSxml(String, String, String, String) : RespuestaDetalleXMLBean[]
+ enviaSMSDetallados(String, SmsDetalladoBean[]) : String
+ enviaSMSDetallados(String, SmsDetalladoBean[], String, String) : String
+ enviaSMSSimples(String, SmsSimpleBean[]) : String
+ enviaSMSSimples(String, SmsSimpleBean[], String, String) : String
+ enviaXML(String, String) : String
+ enviaXML(String, String, String, String) : String
+ EnvioSMS(java.lang.String)
+ getUrlServicio() : String
+ setUrlServicio(String) : void

Para utilizar esta clase es necesario pasarle en el contructor la url del servicio web tal y como se puede ver en el ejemplo que se incluye a continuación:

```
private String URL_SERVICIO=Config.get("ejpl_clientementes_web.ws.sms");

private String enviaSimples() throws AccionException {
        SmsSimpleBean[] mensajes = new SmsSimpleBean[1];
        SmsSimpleBean sms = new SmsSimpleBean();
        String[] telefonos = new String[1];
        telefonos[0] = request.getParameter("TELEFONO");
        sms.setAplicacion(request.getParameter("APLICACION"));
        sms.setConfirmacion(Integer.parseInt(request
                        .getParameter("CONFIRMACION")));
        sms.setExpiracion(1);
        sms.setMensaje(request.getParameter("MENSAJE"));
        sms.setTelefonos(telefonos);
        mensajes[0] = sms;

        EnvioSMS envio = new EnvioSMS(URL_SERVICIO);
        try {
                return envio.enviaSMSSimples("TEST", mensajes);
        } catch (Exception e) {
                throw new AccionException(e.getMessage());
        }
}
```

3.1. CLASES DE PARAMETROS TIPO OBJETOS

3.1.1. SmsSimpleBean

Objeto con los datos básicos para el envío de un sms.

Parámetro	Descripción	Tipo
aplicación	Nombre de la aplicación que realiza el envío. Esta aplicación debe ser previamente autorizada para poder enviar sms.	String
confirmacion	1 si se quiere confirmación y 0 si no se quiere confirmación. No todos los operadores ofrecen confirmación.	int
entregaDiferida	Instante en el que se va a enviar el mensaje (AAAAMMDDHHMM). En blanco si se desea enviar al instante.	String
enviado	Valor fijo a 0	int
expiracion	Tiempo de validez del mensaje en horas de 0 a 23. Si en el tiempo indicado no se ha entregado el mensaje al destinatario entonces no se va a enviar. Cuando el parámetro expiración tiene valor '0' tendrá un tiempo de vida máximo de una semana (168 horas)	int
idPeticion	Identificador de petición. No hace falta rellenarlo para enviar la petición	Long
mensaje	Texto del mensaje a enviar	String
telOrigen	Teléfono que envía el sms (217035)	String
telefonos	Lista con los teléfonos a los que hay que enviar el mensaje	String[]
cdCliente	Código del cliente que solicita el envío (opcional)	String
cdTipoServicio	Código del tipo de servicio (opcional)	String

3.1.2. SmsDetalladoBean

Objeto con los datos básicos para el envío de un sms más los datos de la ventana de tiempo.

Hereda de SmsSimpleBean por lo tanto también tiene los atributos de este objeto.

Parámetro	Descripción	Tipo
aplicación	Nombre de la aplicación que realiza el envío. Esta aplicación debe ser previamente autorizada para poder enviar sms.	String
confirmacion	1 si se quiere confirmación y 0 si no se quiere confirmación. No todos los operadores ofrecen confirmación.	int

entregaDiferida	Instante en el que se va a enviar el mensaje (AAAAMMDDHHMM). En blanco si se desea enviar al instante.	String
enviado	Valor fijo a 0	int
expiracion	Tiempo de validez del mensaje en horas de 0 a 23. Si en el tiempo indicado no se ha entregado el mensaje al destinatario entonces no se va a enviar. Cuando el parámetro expiración tiene valor '0' tendrá un tiempo de vida máximo de una semana (168 horas)	int
idPeticion	Identificador de petición. No hace falta rellenarlo para enviar la petición	Long
mensaje	Texto del mensaje a enviar	String
telOrigen	Teléfono que envía el sms (217035)	String
telefonos	Lista con los teléfonos a los que hay que enviar el mensaje	String[]
cdCliente	Código del cliente que solicita el envío (opcional)	String
cdTipoServicio	Código del tipo de servicio (opcional)	String
clase	Valor fijo a 1	int
horaInicio	Hora de inicio de la ventana horaria a la que se van a enviar los sms. Formato: hh:mm (24 horas)	String
horaFin	Hora de fin de la ventana horaria a la que se van a enviar los sms. Formato: hh:mm (24 horas)	String
lunes	1 indica que se puede enviar en ese día y 0 indica que no	int
martes	1 indica que se puede enviar en ese día y 0 indica que no	int
miercoles	1 indica que se puede enviar en ese día y 0 indica que no	int
jueves	1 indica que se puede enviar en ese día y 0 indica que no	int
viernes	1 indica que se puede enviar en ese día y 0 indica que no	int
sabado	1 indica que se puede enviar en ese día y 0 indica que no	int
domingo	1 indica que se puede enviar en ese día y 0 indica que no	int

3.1.3. ConsultaPeticionBean

Objeto que se rellena con la información de resumen del estado de un envío.

Parámetro	Descripción	Tipo
idPeticion	Identificador de la petición	String
confirmados	Número de sms confirmados por el operador	String
encolados	Número de sms enviados al operador pero que se desconoce su estado ya que hay operadores que no envían confirmación. Por ejemplo Vodafone	String
pendientes	Número de sms pendientes de enviar al operador	String

3.1.4. ConsultaDetalleXMLBean

Bean que se utiliza como parámetro para realizar consultas, según los parámetros de abajo.

Parámetro	Descripción	Tipo
idPeticion	Identificador de petición	String
cd_Cliente	Código de cliente que solicita el envío	String
cd_tiposervicio	Código del tipo de servicio	String
cd_aplicacion	Código del tipo de aplicación	String
fechaenvio	Fecha de envío de los sms	String

3.1.5. RespuestaDetalleXmlBean

Bean que devuelve la situación detallada de los SMS solicitados.

Parámetro	Descripción	Tipo
tDestino	Teléfono de destino	String
texto	Texto del mensaje	String
idPeticion	Identificador de petición	String
fechaSolicitadoEnvio	Fecha de solicitud de envío	String
fechaResultadoEnvio	Fecha de envío	String
horaResultadoEnvio	Hora de envío	String

estadoResultadoEnvio	Estado del envío (Correcto, Teléfono erróneo y Error indefinido: xxxx donde xxxx es el código de error)	String
eConfirma	Estado de la confirmación del envío(Rechazado,Confirmado,Pendiente confirmar,Expirado y no confirmable)	String
fCambioEstado	Fecha en la que se realizó el cambio de estado	String

3.2. METODOS CON PARAMETRO TIPO OBJETO

3.2.1. enviaSMSSimples

Envío de SMS de manera simplificada, sin opciones de ventana de envío. Devuelve un código de petición para su posterior consulta.

Este método está sobrescrito, por lo que es posible invocarlo de las formas siguiente, dependiendo de los parámetros de entrada:

String **enviaSMSSimples**(SmsSimpleBean[] mensajesSMS, String aplicacion)

Parámetros de entrada:

Parámetro	Descripción	Tipo	Obligatorio
mensajesSMS	Array de mensajes a enviar	SmsSimpleBean[]	SI
aplicación	Código de la aplicación que solicita los sms	String	SI

Retorno:

String. Un identificador de petición para poder realizar consultas sobre el estado del envio.

Excepciones:

Exception en el caso de producirse algún error

Ejemplo de uso

Parámetros de entrada:

mensajesSMS:

Parámetro	Descripción
aplicación	“AIAA”
confirmacion	1
entregaDiferida	“200803121000”
enviado	0
expiracion	1
idPeticion	null
mensaje	“Prueba”
telOrigen	“217035”

telefonos	"XXXXXXXXX"

Aplicación:
"AIAA"

Retorno:

"1407" – Identificador de petición para poder realizar consultas sobre el estado del envio. Se genera un identificador distinto para cada petición.

Nota	*El código de aplicación AIAA es un código de pruebas. En producción este código no existe y cada aplicación debe solicitar su propio código de aplicación.*

String **enviaSMSSimples**(SmsSimpleBean[] mensajesSMS, String aplicación, String cliente, String tipoServicio)

Se incluye la posibilidad de llamada a enviaSMSSimple, con dos nuevos campos (cliente y tipoServicio), pensados para aquellos aplicativos que desean organizar sus envíos en base a quienes son los clientes de los mismos y/o el tipo de servicio que utilizan.

Parámetros de entrada:

Parámetro	Descripción	Tipo	Obligatorio
mensajesSMS	Array de mensajes a enviar	SmsSimpleBean[]	SI
aplicación	Código de la aplicación que solicita los sms	String	SI
cliente	Identificador del cliente de la aplicación que realiza el envío.	String	SI
tipoServicio	Identificador de tipo de servicio de la aplicación que realiza el envío.	String	SI

Retorno:

String. Un identificador de petición para poder realizar consultas sobre el estado del envio.

Excepciones:
Exception en el caso de producirse algún error

3.2.2. enviaSMSDetallados

Envía los mensajes SMS que recibe en el SmsDetalladoBean en la ventana de tiempo indicada en el objeto.

Este método está sobrescrito, por lo que es posible invocarlo de las formas siguiente, dependiendo de los parámetros de entrada:

String **enviaSMSDetallados**(SmsDetalladoBean[] mensajesSMS, String aplicación)

Parámetros de entrada:

Parámetro	Descripción	Tipo	Obligatorio
mensajesSMS	Mensajes sms que se desea enviar	SmsDetalladoBean[]	SI
aplicación	Código de la aplicación que solicita los sms	String	SI

Retorno:

String. Un identificador de petición para poder realizar consultas sobre el estado del envío.

Excepciones:

Exception en el caso de producirse algún error.

Ejemplo de uso

Parámetros de entrada:

MensajesSMS:

Parámetro	Descripción
aplicación	“AIAA”
confirmacion	1
entregaDiferida	“200803171000”
enviado	0
expiracion	1
idPeticion	null
mensaje	“Mensaje a enviar”
telOrigen	“217035”
telefonos	“XXXXXXXXX”
Clase	1
horaFin	“10:00”
horaInicio	“09:00”
lunes	1
martes	1
miercoles	1
jueves	1

viernes	1
sabado	1
domingo	1

Aplicación:
"AIAA"

Retorno:

"1408" – Identificador de petición para poder realizar consultas sobre el estado del envio. Se genera un identificador distinto para cada petición.

String **enviaSMSDetallados**(SmsDetalladoBean[] mensajesSMS, String aplicación, String cliente, String tipoServicio)

Se incluye la posibilidad de llamada a enviaSMSDetallados, con dos nuevos campos (cliente y tipoServicio), pensados para aquellos aplicativos que desean organizar sus envíos en base a quienes son los clientes de los mismos y/o el tipo de servicio que utilizan.

Parámetros de entrada:

Parámetro	Descripción	Tipo	Obligatorio
mensajesSMS	Mensajes sms que se desea enviar	SmsDetalladoBean[]	SI
aplicación	Código de la aplicación que solicita los sms	String	SI
cliente	Identificador del cliente de la aplicación que realiza el envío.	String	SI
tipoServicio	Identificador de tipo de servicio de la aplicación que realiza el envío.	String	SI

Retorno:

String. Un identificador de petición para poder realizar consultas sobre el estado del envío.

Excepciones:
Exception en el caso de producirse algún error.

3.2.3. consultaPeticion

Consulta el estado de confirmación de los sms de una petición.

ConsultaPeticionBean **consultaPeticion**(String idPeticion)

Parámetros de entrada:

Parámetro	Descripción	Tipo	Obligatorio
idPeticion	Identificador de la petición de envio. Obtenido de cualquiera de los tres metodos de envio	String	SI

Retorno:

ConsultaPeticionBean – Numero de sms confirmados, encolados o pendientes.

Excepciones:

Exception – En el caso de producirse alguna excepción

Ejemplo de uso

Parámetros de entrada:

idPeticion:
"1408"

Retorno:

ConsultaPeticionBean:

Parámetro	Descripción
idPeticion	"1408"
confirmados	"1"
encolados	"0"
pendientes	"0"

3.2.1. consultaDetallePeticion

Consulta la situación detallada de cada uno de los SMS solicitados para una petición concreta.

RespuestaDetalleXMLBean[] **consultaDetallePeticion** (String idPeticion, String aplicacion)

Parámetros de entrada:

Parámetro	Descripción	Tipo	Obligatorio
idPeticion	Identificador de la petición de envio. Obtenido de cualquiera de los tres metodos de envio	String	SI

aplicacion	Código de la aplicación que solicita los sms	String	SI

Retorno:

RespuestaDetalleXMLBean[]– Array de RespuestaDetalleXMLBean que contienen el detalle de los SMS solicitados para la petición concreta.

Excepciones:

Exception – En el caso de producirse alguna excepción

Ejemplo de uso

Parámetros de entrada:

idPeticion:
"1408"

aplicacion:
"AIAA"

Retorno:

RespuestaDetalleXMLBean[]: Donde cada RespuestaDetalleXMLBean tendrá los siguientes campos:

Parámetro	Descripción
idPeticion	"1408"
tDestino	"6XXXXXXXX"
texto	"Mensaje enviado"
fechaSolicitadoEnvio	"08-07-2011"
fechaResultadodEnvio	"09-07-2011"
horaResultadoEnvio	"09:00"
fCambioEstado	"09-07-2011"
estadoResultadodEnvio	"Envio Correcto"
eConfirma	"CONFIRMADO"

3.2.2. consultaDetallePeticionTime

Consulta la situación detallada de cada uno de los SMS solicitados para una petición concreta. Este método devuelve la fecha de cambio incluyendo horas minutos y segundos.

RespuestaDetalleXMLBean[] **consultaDetallePeticion** (String idPeticion, String aplicacion)

Parámetros de entrada:

Parámetro	Descripción	Tipo	Obligatorio
idPeticion	Identificador de la petición de envio. Obtenido de cualquiera de los tres metodos de envio	String	SI
aplicacion	Código de la aplicación que solicita los sms	String	SI

Retorno:

RespuestaDetalleXMLBean[]– Array de RespuestaDetalleXMLBean que contienen el detalle de los SMS solicitados para la petición concreta.

Excepciones:

Exception – En el caso de producirse alguna excepción

Ejemplo de uso

Parámetros de entrada:

idPeticion:
"1408"

aplicacion:
"AIAA"

Retorno:

RespuestaDetalleXMLBean[]: Donde cada RespuestaDetalleXMLBean tendrá los siguientes campos:

Parámetro	Descripción
idPeticion	"1408"
tDestino	"6XXXXXXXX"
texto	"Mensaje enviado"
fechaSolicitadoEnvio	"08-07-2011"
fechaResultadodEnvio	"09-07-2011"
horaResultadoEnvio	"09:00"
fCambioEstado	"09-07-2011 10:10:10"
estadoResultadodEnvio	"Envio Correcto"
eConfirma	"CONFIRMADO"

3.2.3. consultaSMSxml

Consulta la situación detallada de cada uno de los SMS solicitados en una fecha concreta para una aplicación, o aplicación, centro y servicio.

Este método es posible invocarlo con diferente número de parámetros:

RespuestaDetalleXMLBean[] **consultaSMSxml** (String aplicacion, String fechaEnvio)

Consulta la situación detallada de cada uno de los SMS solicitados en una fecha concreta para una aplicación

Parámetros de entrada:

Parámetro	Descripción	Tipo	Obligatorio
aplicacion	Código de la aplicación que solicita los sms	String	SI
fechaEnvio	Fecha en la que se ha producido el envío	String	SI

Retorno:

RespuestaDetalleXMLBean[]– Array de RespuestaDetalleXMLBean que contienen el detalle de los SMS solicitados para la petición concreta.

Excepciones:

Exception – En el caso de producirse alguna excepción

Ejemplo de uso

Parámetros de entrada:

aplicacion:
"AIAA"

fechaEnvio:
"14-06-2011"

Retorno:

RespuestaDetalleXMLBean[]: Donde cada RespuestaDetalleXMLBean tendrá los siguientes campos:

Parámetro	Descripción
idPeticion	"32"
tDestino	"6XXXXXXXX"
texto	"Mensaje a consultar"
fechaSolicitadoEnvio	"14-06-2011"
fechaResultadodEnvio	"14-06-2011"

horaResultadoEnvio	“12:00”
fCambioEstado	“14-06-2011”
estadoResultadodEnvio	“Teléfono erróneo”
eConfirma	“RECHAZADO”

RespuestaDetalleXMLBean[] **consultaSMSxml** (String aplicacion, String fechaEnvio, String tipoServicio, String codCentro)

Consulta la situación detallada de cada uno de los SMS solicitados en una fecha concreta para una aplicación, centro y servicio.

Parámetros de entrada:

Parámetro	Descripción	Tipo	Obligatorio
aplicacion	Código de la aplicación que solicita los sms	String	SI
fechaEnvio	Fecha en la que se ha producido el envío	String	SI
tipoServicio	Identificador de tipo de servicio de la aplicación que realiza el envío.	String	SI
codCentro	Identificador del cliente de la aplicación que realiza el envío.	String	SI

Retorno:

RespuestaDetalleXMLBean[]– Array de RespuestaDetalleXMLBean que contienen el detalle de los SMS solicitados para la petición concreta.

Excepciones:

Exception – En el caso de producirse alguna excepción

Ejemplo de uso

Parámetros de entrada:

aplicacion:
“AIAA”

fechaEnvio:
“08-07-2011”

tipoServicio:
“tipoServicio”

codCentro:
“cliente”

Retorno:

RespuestaDetalleXMLBean[]: Donde cada RespuestaDetalleXMLBean tendrá los siguientes campos:

Parámetro	Descripción
idPeticion	“80”
tDestino	“636959744”
texto	“Pruebas Envío XXXXX realizado.”
fechaSolicitadoEnvio	“08-07-2011”
fechaResultadodEnvio	“11-10-2011”
horaResultadoEnvio	“13:20:06”
fCambioEstado	“11-10-2011”
estadoResultadodEnvio	“Envio Correcto”
eConfirma	“RECHAZADO”

3.1. ESQUEMAS PARA LOS FICHEROS XML

3.1.1. XML de envío

El xml de envío se debe componer de acuerdo al siguiente esquema (envio.xsd):

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
    <xs:element name="MENSAJE" type="xs:string"/>
    <xs:element name="PETICION">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="SMS" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="SMS">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="MENSAJE"/>
                <xs:element ref="TELEFONO" maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:attribute name="confirmacion" type="xs:byte" use="optional" default="0"/>
            <xs:attribute name="entregaDiferida" type="xs:string" use="optional"/>
            <xs:attribute name="expiracion" type="xs:byte" use="optional" default="0"/>
            <xs:attribute name="telOrigen" type="xs:string" use="required"/>
            <xs:attribute name="lunes" type="xs:byte" use="optional" default="1"/>
            <xs:attribute name="martes" type="xs:byte" use="optional" default="1"/>
            <xs:attribute name="miercoles" type="xs:byte" use="optional" default="1"/>
            <xs:attribute name="jueves" type="xs:byte" use="optional" default="1"/>
            <xs:attribute name="viernes" type="xs:byte" use="optional" default="1"/>
            <xs:attribute name="sabado" type="xs:byte" use="optional" default="1"/>
            <xs:attribute name="domingo" type="xs:byte" use="optional" default="1"/>
            <xs:attribute name="horaInicio" type="xs:string" use="optional" default="00:00"/>
            <xs:attribute name="horaFin" type="xs:string" use="optional" default="23:59"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="TELEFONO">
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:minLength value="9"/>
                <xs:maxLength value="9"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:element>
</xs:schema>
```

Ejemplo de XML de envío

```
<?xml version="1.0" encoding="UTF-8"?>
<PETICION>
    <SMS confirmacion="0" expiracion="0" telOrigen="217035" lunes="1" martes="1" miercoles="1" jueves="1"
viernes="1" sabado="1" domingo="1" horaInicio="00:00" horaFin="23:59">
        <MENSAJE>prueba de envio</MENSAJE>
        <TELEFONO>XXXXXXXXX</TELEFONO>
        <TELEFONO>XXXXXXXXX</TELEFONO>
    </SMS>
    <SMS confirmacion="1" expiracion="0" telOrigen="217035" lunes="1" martes="1" miercoles="1" jueves="1"
viernes="1" sabado="1" domingo="1" horaInicio="00:00" horaFin="23:59">
        <MENSAJE>Segunda prueba envio</MENSAJE>
        <TELEFONO>XXXXXXXXX</TELEFONO>
    </SMS>
</PETICION>
```

3.1.2. XML de consulta

El xml de consulta que nos devuelve en el método consultaPeticionForms es de acuerdo al siguiente esquema (consulta.xsd):

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
    <xs:element name="confirmados" type="xs:int">
        <xs:annotation>
            <xs:documentation>Número de sms confirmados</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element name="encolados" type="xs:int">
        <xs:annotation>
            <xs:documentation>Número de sms encolados</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element name="pendientes" type="xs:int">
        <xs:annotation>
            <xs:documentation>Número de sms pendientes</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element name="consultaPeticion">
        <xs:complexType mixed="true">
            <xs:sequence>
                <xs:element ref="confirmados"/>
                <xs:element ref="encolados"/>
                <xs:element ref="pendientes"/>
            </xs:sequence>
            <xs:attribute name="idPeticion" type="xs:short" use="required"/>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

Ejemplo de XML de respuesta de la consulta:

```
<?xml version="1.0" encoding="UTF-8"?>
<consultaPeticion idPeticion="1408">
    <confirmados>2</confirmados>
    <encolados>0</encolados>
    <pendientes>0</pendientes>
</consultaPeticion>
```

3.2. METODOS CON PARAMETRO TIPO STRING QUE CONTIENE UN XML

3.2.1. enviaForms

Envío de SMS utilizando xml como parámetros de entrada/salida.

String **enviaForms**(String aplicacion, String mensajes)

Parámetros de entrada:

Parámetro	Descripción	Tipo	Obligatorio
aplicacion	Código de la aplicación que solicita los sms	String	SI
mensajes	XML donde se definen los sms que se desean enviar	String (Schema envio.xsd)	SI

Retorno:

String. Un identificador de petición para poder realizar consultas sobre el estado del envio

Excepciones:

Exception en el caso de producirse algún error
XmlException en el caso de que el formato del XML no sea correcto

Ejemplo de uso

Parámetros de entrada:

Aplicación:
"AIAA"

Mensajes:

```
<PETICION>
    <SMS confirmacion="0" expiracion="0" telOrigen="217035" lunes="1" martes="1" miercoles="1" jueves="1"
viernes="1" sabado="1" domingo="1" horaInicio="00:00" horaFin="23:59">
        <MENSAJE>prueba de envioXml</MENSAJE>
        <TELEFONO>XXXXXXXXX</TELEFONO>
    </SMS>
    <SMS confirmacion="1" expiracion="0" telOrigen="217035" lunes="1" martes="1" miercoles="1" jueves="1"
viernes="1" sabado="1" domingo="1" horaInicio="00:00" horaFin="23:59">
        <MENSAJE>Segunda prueba envio xml</MENSAJE>
        <TELEFONO>XXXXXXXXX</TELEFONO>
    </SMS>
</PETICION>
```

Retorno:

"1409" – Identificador de petición para poder realizar consultas sobre el estado del envio. Se genera un identificador distinto para cada petición.

3.2.2. enviaSMSxml

Envío de SMS utilizando xml como parámetros de entrada/salida.

String **enviaSMSxml**(String aplicacion, String mensajes, String cliente, String tipoServicio)

Parámetros de entrada:

Parámetro	Descripción	Tipo	Obligatorio
aplicacion	Código de la aplicación que solicita los sms	String	SI
mensajes	XML donde se definen los sms que se desean enviar	String (Schema envio.xsd)	SI
cliente	Identificador del cliente de la aplicación que realiza el envío.	String	SI
tipoServicio	Identificador de tipo de servicio de la aplicación que realiza el envío.	String	SI

Retorno:

String. Un identificador de petición para poder realizar consultas sobre el estado del envio

Excepciones:

Exception en el caso de producirse algún error
XmlException en el caso de que el formato del XML no sea correcto

Ejemplo de uso

Parámetros de entrada:

Aplicación:
"AIAA"
Cliente:
"Sanidad"
tipoServicio:
"HospPAsturias"
Mensajes:

```
<PETICION>
    <SMS confirmacion="0" expiracion="0" telOrigen="217035" lunes="1" martes="1" miercoles="1" jueves="1"
viernes="1" sabado="1" domingo="1" horaInicio="00:00" horaFin="23:59">
        <MENSAJE>prueba de envioXml</MENSAJE>
        <TELEFONO>XXXXXXXXX</TELEFONO>
    </SMS>
    <SMS confirmacion="1" expiracion="0" telOrigen="217035" lunes="1" martes="1" miercoles="1" jueves="1"
viernes="1" sabado="1" domingo="1" horaInicio="00:00" horaFin="23:59">
        <MENSAJE>Segunda prueba envio xml</MENSAJE>
        <TELEFONO>XXXXXXXXX</TELEFONO>
    </SMS>
</PETICION>
```

Retorno:

"1409" – Identificador de petición para poder realizar consultas sobre el estado del envio. Se genera un identificador distinto para cada petición.

3.2.3. consultaPeticionForms

Método de consulta de estado de una petición. Devuelve un string con el estado de los SMS solicitados con el siguiente formato: C=nº confirmados;P=nº pendientes;E=nº encolados.

String **consultaPeticionForms**(String idPeticion)

Parámetros de entrada:

Parámetro	Descripción	Tipo	Obligatorio
idPeticion	Identificador de la petición de envio. Obtenido de cualquiera de los tres metodos de envio	String	SI

Retorno:

String - XML con la respuesta de la consulta (Schema consulta.xsd)

Excepciones:

Exception – en el caso de producirse un error

Ejemplo de uso

Parámetros de entrada:

idPeticion:
"1408"

Retorno:

```
<?xml version="1.0" encoding="UTF-8"?>
<consultaPeticion idPeticion="1408">
        <confirmados>2</confirmados>
        <encolados>0</encolados>
        <pendientes>0</pendientes>
</consultaPeticion>
```

3.2.4. consultaDetallePeticionForms

Método de consulta,devuelve la situación detallada de cada uno de los SMS solicitados para una petición concreta,y una aplicación concreta,en formato xml

String **consultaDetallePeticionForms** (String idPeticion,String aplicacion)

Parámetros de entrada:

Parámetro	Descripción	Tipo	Obligatorio
idPeticion	Identificador de la petición de envío.	String	SI
aplicacion	Código de la aplicación que solicita los sms	String	SI

Retorno:

String que contiene un xml con los datos de la petición

Excepciones:

Exception – en el caso de producirse un error

Ejemplo de uso

Parámetros de entrada:

idPeticion:
"20"
aplicación:
"AIAA"

Retorno:

<respuestaDetalle><sms><correcto>false</correcto><eConfirma>null</eConfirma><estadoResultadodEnvio>Telefono erroneo</estadoResultadodEnvio><fCambioEstado>11-10-2011</fCambioEstado><fechaResultadodEnvio>11-10-2011</fechaResultadodEnvio><fechaSolicitadoEnvio>14-06-2011</fechaSolicitadoEnvio><horaResultadoEnvio>13:20:06</horaResultadoEnvio><idPeticion>20</idPeticion><tDestino>61XXXXXXX</tDestino><texto>Envio especificando 217035</texto></sms></respuestaDetalle>

3.2.5. consultaSMSxmlForms

Consulta la situación detallada de cada uno de los SMS solicitados en una fecha concreta.

String **consultaSMSxmlForms** (String aplicacion,String fechaEnvio)

Parámetros de entrada:

Parámetro	Descripción	Tipo	Obligatorio
aplicacion	Código de aplicación.	String	SI
fechaEnvio	Fecha de Envío de los sms	String	SI

Retorno:

String: xml que contienen el detalle de los SMS solicitados para la petición concreta

Excepciones:

Exception – en el caso de producirse un error

Ejemplo de uso

Parámetros de entrada:

aplicacion:
"AIAA"
fechaEnvio:
"27-10-2011"

Retorno:

<respuestaDetalle><sms><correcto>false</correcto><eConfirma>null</eConfirma><estadoResultadodEnvio>Envio Correcto</estadoResultadodEnvio><fCambioEstado>27-10-2011</fCambioEstado><fechaResultadodEnvio>27-10-2011</fechaResultadodEnvio><fechaSolicitadoEnvio>27-10-2011</fechaSolicitadoEnvio><horaResultadoEnvio>11:50:45</horaResultadoEnvio><idPeticion>21072</idPeticion><tDestino>654927559</tDestino><texto>PRUEBA DEL SERVICIO DESDE HRC</texto></sms><sms><correcto>false</correcto><eConfirma>null</eConfirma><estadoResultadodEnvio>Envio Correcto</estadoResultadodEnvio><fCambioEstado>27-10-2011</fCambioEstado><fechaResultadodEnvio>27-10-2011</fechaResultadodEnvio><fechaSolicitadoEnvio>27-10-2011</fechaSolicitadoEnvio><horaResultadoEnvio>12:01:47</horaResultadoEnvio><idPeticion>21073</idPeticion><tDestino>654927559</tDestino><texto>PRUEBA3 DEL SERVICIO DESDE HRC</texto></sms><sms><correcto>false</correcto><eConfirma>null</eConfirma><estadoResultadodEnvio>Envio Correcto</estadoResultadodEnvio><fCambioEstado>27-10-2011</fCambioEstado><fechaResultadodEnvio>27-10-2011</fechaResultadodEnvio><fechaSolicitadoEnvio>27-10-2011</fechaSolicitadoEnvio><horaResultadoEnvio>16:49:45</horaResultadoEnvio><idPeticion>21078</idPeticion><tDestino>666666660</tDestino><texto>Este texto contiene la longitud exacta de 480 caracteres: 123456789-123456789-123456789-

Exception – en el caso de producirse un error

Ejemplo de uso

Parámetros de entrada:

aplicacion:
"AIAA"
fechaEnvio:
"08-07-2011"
tipoServicio:
"tipoServicio"
codCentro:
"cliente"

Retorno:

<respuestaDetalle><sms><correcto>false</correcto><eConfirma>CONFIRMADO</eConfirma><estadoResultadodEnvio>Envio Correcto</estadoResultadodEnvio><fCambioEstado>11-10-2011</fCambioEstado><fechaResultadodEnvio>11-10-2011</fechaResultadodEnvio><fechaSolicitadoEnvio>08-07-2011</fechaSolicitadoEnvio><horaResultadoEnvio>13:20:06</horaResultadoEnvio><idPeticion>80</idPeticion><tDestino>636959744</tDestino><texto>Pruebas Envío XXXXX realizado. </texto></sms></respuestaDetalle>

4. URL DE ACCESO AL SERVICIO WEB

La url de acceso al servicio web depende del entorno en el que estemos trabajando.

4.1. Desarrollo

En el entorno de desarrollo la url del servicio es la siguiente:

http://deswebservices.madrid.org/mentes_ws

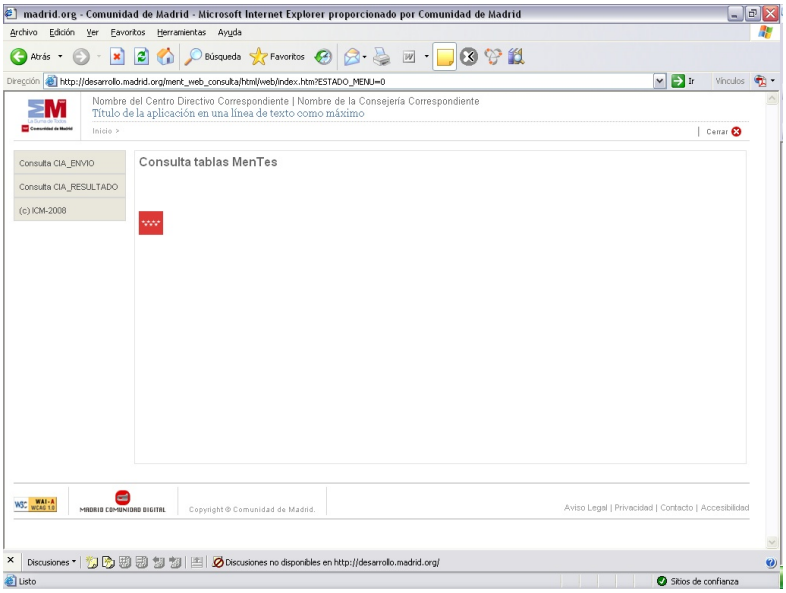
En el entorno de desarrollo no se realiza el envío de sms real sino que simplemente se introduce el mensaje enviado en una tabla de Oracle.

Para comprobar que los mensajes se han escrito en la tabla de Oracle existe una aplicación en el entorno de desarrollo que nos permite hacer una consulta sobre dicha tabla.

La url de la aplicación de consulta de entradas de mensajes en desarrollo es:

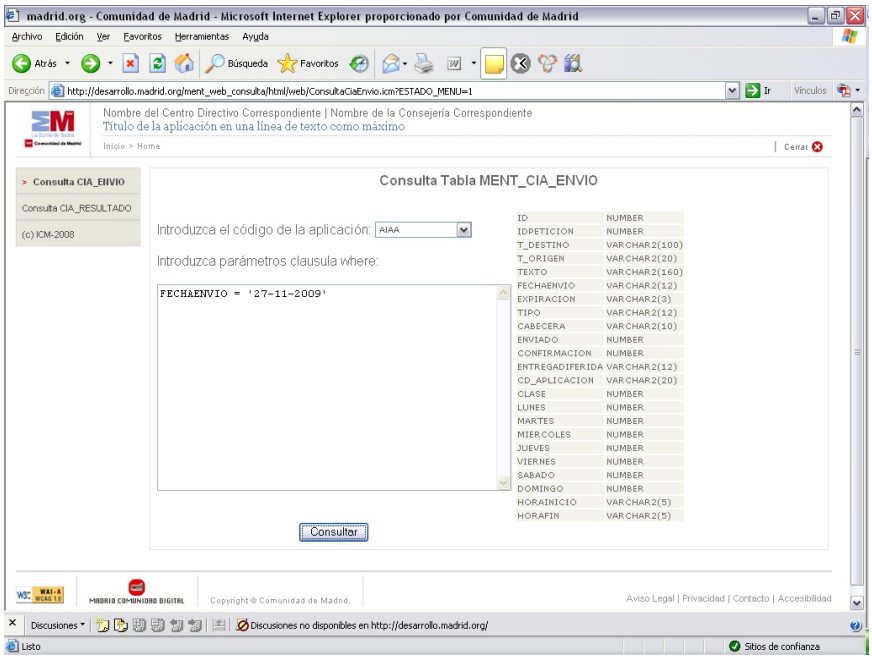
http://desarrollo.madrid.org/ment_web_consulta

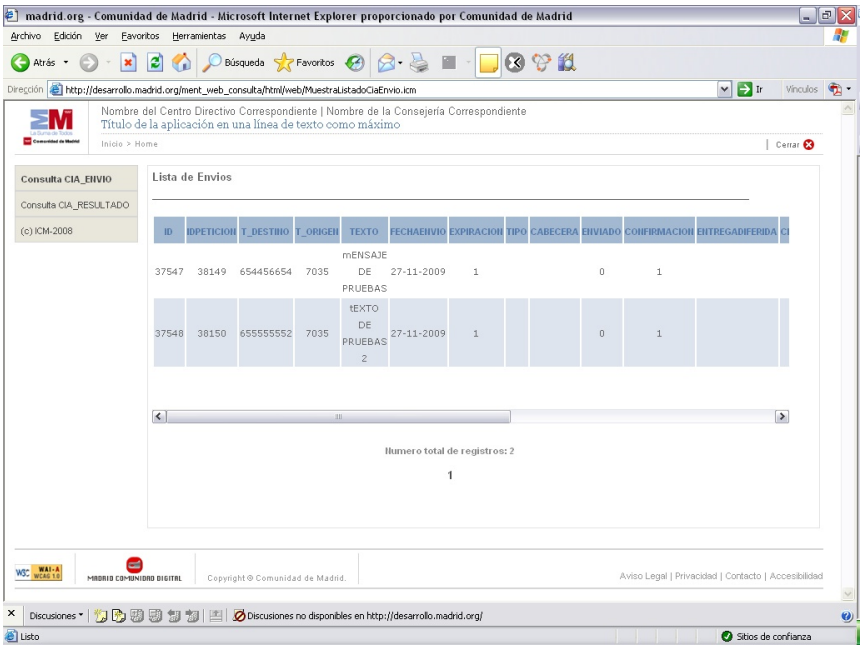
Accediendo a esta url nos pedirá usuario y password y debemos introducir ADMON/ADMON. A continuación nos aparece la siguiente pantalla:


madrid.org - Comunidad de Madrid - Microsoft Internet Explorer proporcionado por Comunidad de Madrid
Archivo Edición Ver Favoritos Herramientas Ayuda
http://desarrollo.madrid.org/ment_web_consulta/html/web/index.htm?ESTADO_MENU=0
Nombre del Centro Directivo Correspondiente | Nombre de la Consejería Correspondiente
Título de la aplicación en una línea de texto como máximo
Consulta CIA_ENVIO
Consulta CIA_RESULTADO
(c) ICM-2008
Consulta tablas MenTes
Aviso Legal | Privacidad | Contacto | Accesibilidad
Discusiones no disponibles en http://desarrollo.madrid.org/
Listo
Sitios de confianza

Pulsando sobre la opción **Consulta CIA_ENVIO** nos sale un formulario que nos permite introducir el código de aplicación que tenemos asignado y filtrar por lo campos que nos interese para localizar los mensajes enviados. Como ayuda se presenta a la derecha la lista de los campos de la tabla.

A continuación se muestra un ejemplo de consulta:




madrid.org - Comunidad de Madrid - Microsoft Internet Explorer proporcionado por Comunidad de Madrid
Nombre del Centro Directivo Correspondiente | Nombre de la Consejería Correspondiente
Título de la aplicación en una línea de texto como máximo
Consulta CIA_ENVIO
Consulta CIA_RESULTADO
(c) ICM-2008
Lista de Envios
ID IDPETICION T_DESTINO T_ORIGEN TEXTO FECHAENVIO EXPIRACION TIPO CABECERA ENVIADO CONFIRMACION ENTREGADIFERIDA
37547 38149 654456654 7035 mENSAJE DE PRUEBAS 27-11-2009 1 0 1
37548 38150 655555552 7035 tEXTO DE PRUEBAS 2 27-11-2009 1 0 1
Numero total de registros: 2
1

Nota: En el caso de que se desee filtrar por varios campos estos se separan como el formato de SQL (AND, OR, etc)

5. METODOS Y PARAMETROS DEPRECADOS

5.1. EnvioBean

Bean que se rellena en la consulta ConsultaEnvios por la aplicación y la fecha de envío.

Parámetro	Descripción	Tipo
nm_mensajes_env	Número de mensajes enviados	int

nm_mensajes_sol	Número de mensajes solicitados	int
aplicacion	Código del tipo de aplicación	String
fecha	Fecha de envío de los sms	String

5.2. ListaEnviosBean

Lista de Bean del tipo EnvioBean, que se devuelve en la consulta ConsultaEnvios realizadas por aplicación, o por aplicación y fecha.

Parámetro	Descripción	Tipo
listaEnvios	Array de EnvioBean	EnvioBean[]

5.3. RespuestaBean

Bean que devuelve el número de SMS enviados, y una lista que puede contener el código del error o el identificador de petición.

Parámetro	Descripción	Tipo
numEnviados	Número de mensajes enviados	int
errores	Puede almacenar tanto -1 si es un error, o el identificador de petición para poder realizar consultas sobre el estado del envío.	String[]

5.4. SmsBean

Bean que se usa para el envío de SMS

Parámetro	Descripción	Tipo
mensaje	Contenido del mensaje	String
telefonos	Lista de teléfonos	String[]

5.5. ConsultaEnvios

Método deprecated,se mantiene por compatibilidad con los clientes del webservice ssms_ws

ListaEnviosBean **ConsultaEnvios**(EnvioBean envio)

Parámetros de entrada:

Parámetro	Descripción	Tipo	Obligatorio
envio	Fecha ,código de aplicación, Numero de mensajes solicitados y enviados	EnvioBean	SI

Retorno:

ListaEnviosBean.

Excepciones:

Exception en el caso de producirse algún error

Ejemplo de uso

Parámetros de entrada:

EnvioBean:

Parámetro	Descripción
aplicación	“AIAA”
fecha	“20/09/2011”
nm_mensajes_env	1
nm_mensajes_sol	1

Retorno:

ListaEnviosBean

5.6. EnviaMensajes

Método deprecated,se mantiene por compatibilidad con los clientes del webservice ssms_ws

RespuestaBean **EnviaMensajes**(SmsBean[] mensajesSMS,String aplicacion)

Parámetros de entrada:

Parámetro	Descripción	Tipo	Obligatorio
mensajesSMS	Array de Mensajes a enviar	SmsBean[]	SI
Aplicación	Código de la aplicación	String	SI

Retorno:

RespuestaBean.

En la clase RespuestaBean, devuelve un código de error -1,en los mismos casos que en el método enviaSMSSimples comentado en el punto 2.1,ya que utiliza el mismo método para enviar, previa transformación de los objetos SmsBean[] a SmsSimpleBean[]

Excepciones:

Exception en el caso de producirse algún error

Ejemplo de uso

Parámetros de entrada:

SmSBean[0]:

Parámetro	Descripción
mensaje	“Esto es una prueba”
Telefonos[0]	“666666660”

Retorno:

RespuestaBean

Parámetro	Descripción
numEnviados	1
errores[0]	“21078”

6. PREGUNTAS MÁS FRECUENTES

En este apartado se incluyen una lista de preguntas más frecuentes sobre el servicio

Pregunta: ¿Estoy probando el servicio web contra el entorno de desarrollo y no recibo ningún mensaje?

Respuesta: El entorno de desarrollo se llega a realizar el envío de mensajes.

Pregunta: ¿Al invocar al servicio web me dice la aplicación XXXX no está autorizada?

Respuesta: Para poder acceder al servicio web es necesario solicitar previamente un código de aplicación.